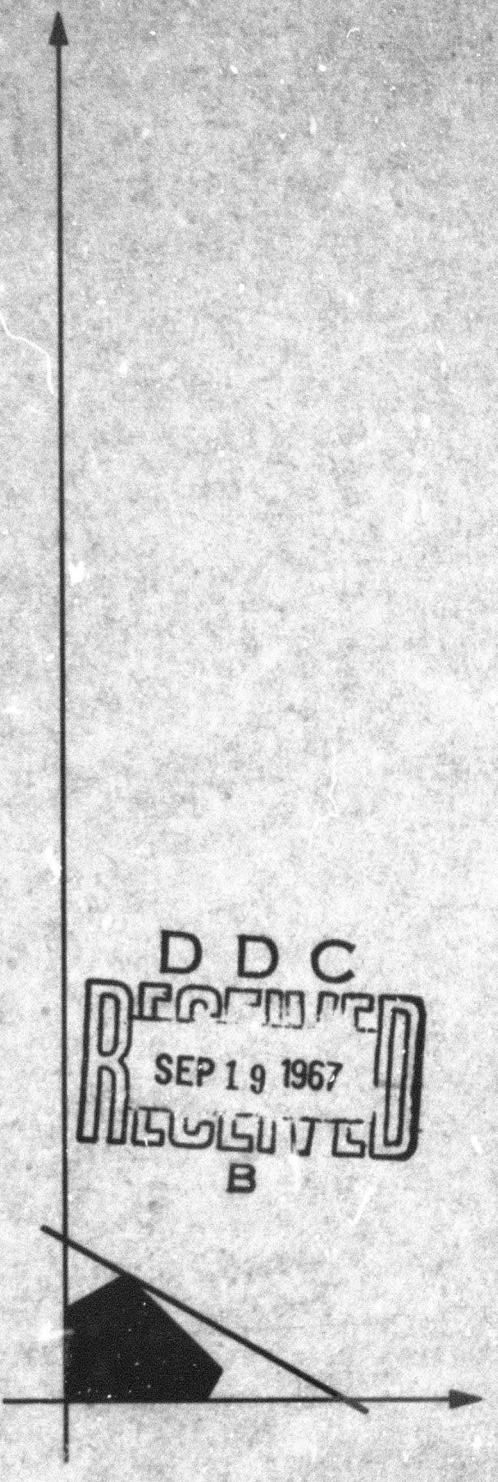
AD658065

ORC 67-26
JULY 1967

HOW TO SOLVE LINEAR INEQUALITIES

by

David Gale



DDC
RECEIVED
SEP 19 1967
B

OPERATIONS RESEARCH CENTER

COLLEGE OF ENGINEERING

This document has been approved for public release and sale; its distribution is unlimited.

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va 22151

UNIVERSITY OF CALIFORNIA-BERKELEY

HOW TO SOLVE LINEAR INEQUALITIES

by

David Gale
Operations Research Center
University of California, Berkeley

July 1967 ORC 67-26

This research has been partially supported by the Office of Naval Research under Contract Nonr-222(83), and the National Science Foundation under Grant GP-7417 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

ABSTRACT

This paper is an attempt to solve the following constrained minimum problem: To present the most easily described algorithm for solving linear inequalities subject to the constraints,

(1) The algorithm must be efficient.
(2) It must be shown to terminate.

The algorithm is a variant of the lexicographic simplex method which avoids using any artificial objective function.

HOW TO SOLVE LINEAR INEQUALITIES

by

David Gale

1. Introduction

Suppose you were asked to solve a system of simultaneous linear inequalities of modest size, say for example, four inequalities in three unknowns. How would you proceed? Or suppose that the size of the problem was immodest so that machine computation was appropriate? How should the machine proceed?

These questions, it seems to me, are natural ones to ask, for linear inequalities come up almost as often as linear equations in all sorts of applications; yet I believe very few mathematicians can give a good answer to them. I suspect, given a little time, a competent mathematician could devise some sort of finite algorithm which for any system of inequalities would either produce a solution or else show that none existed. It would be surprising, though, if he could on the spur of the moment come up with a procedure that would do the job using only a "reasonable" amount of computation. By a reasonable amount of computation I mean an amount of the same order of magnitude as that involved in solving systems of equations. In fact, this raises a mathematical question. Do there exist such reasonable procedures, or is the inequality problem intrinscally of a higher order of computational complexity then the equation problem? I would like to expound briefly on the present curious state of affairs regarding this question.

The usual method for solving linear equations is ordinary "elimination;" solve equation 1 for x_1 and then substitute this expression into equations 2 through m, etc. In this method the basic step is this elimination, and after each of the n variables have been eliminated, thus after n such steps, the solution emerges

(this description is not intended to be precise or rigorous). The algorithm we are about to describe makes use of these same elimination or, as we shall call them, replacement steps. The number of such steps will in general be greater than n, but not much greater, perhaps as much as $2n$. Klee [5] has constructed examples which indicate that one may run into situations which require as much as (roughly) mn steps and has conjectured that this is the maximum possible. On the other hand, the best upper bound on the number of steps which can be rigorously established is not even algebraic in m and n but of the order of $\binom{n}{m}$. Thus there is a large and embarrassing gap between what has been observed and what has been proved. This gap has stood as a challenge to workers in the field for twenty years now and remains, in my opinion, the principal open question in the theory of linear computation.

One further introductory word seems in order. There is one group in the mathematical community who do know how to solve inequalities and these are the people who work in linear programming. The situation here is again curious. Linear programming involves maximizing or minimizing a linear function using variables which are required to satisfy a system of linear inequalites. Thus, in order to solve a linear program one must in the process find a solution of these inequalities. It turns out, on the other hand, that the problem of solving inequalities can itself be thought of as a linear programming problem in which one is minimizing a so-called "artificial objective function." While this approach achieves the desired end, it seems to me to be a backward way of going about things. Logically one would first learn to solve the inequalities and then worry about minimizing or maximizing over the set of solutions. This is the approach taken here. The method used in the lexicographic variant of the simplex method of Dantzig, Orden, and Wolfe [1] which was used by those authors to solve linear

programs and later by Dantzig [2] to solve matrix games, but has not up to now, to my knowledge, been used to give a direct method (no artificial objective function) for solving inequalities. A different direct method has been given by Debreu [3] but his procedure is more complicated to describe than the one proposed here, though it may be computationally more efficient in some cases.

2. Solving Matrix Equations

We begin by reviewing the "standard" method for solving linear equations, slightly generalized and using slightly different terminology from the usual one.

PROBLEM I: Given an $m \times n$ matrix A and an $m \times r$ matrix B, find an $n \times r$ matrix Y such that $AY = B$.

It will be convenient to rephrase the problem. Instead of thinking of A and B as matrices we will think of them as sets of m-vectors. Thus

$$A = \{a_1, \ldots, a_n\}$$
$$B = \{b_1, \ldots, b_r\}$$

Problem: Express each vector b_k as a linear combination of the vectors a_j if possible.

We are about to describe what we will call a replacement algorithm for solving (I). The following is the fundamental notion needed.

DEFINITION: Let $S = \{s_1, \ldots, s_m\}$ be a basis for m-space and let $B = \{b_1, \ldots, b_r\}$ be any set of m-vectors. The tableau of B with respect to S is the $m \times r$ matrix $Y = (y_{ij})$ such that

$$b_j = \sum_{i=1}^{m} y_{ij} a_i, \quad j = 1, \ldots, r .$$

In matrix notation, if we think of S and B as matrices with columns s_i and b_j then Y is simply the solution if the equation $SY = B$, or $Y = S^{-1}B$.

We write tableaus in the following manner:

	b_1	b_2	...	b_n
s_1	y_{11}	y_{12}	...	y_{1n}
s_2	y_{21}	y_{22}	...	y_{2n}
$\vdots$				
s_m	y_{m1}	y_{m2}	...	y_{mn}

Figure 1.

REPLACEMENT ALGORITHM: We are going to describe a procedure for constructing a finite sequence of bases. The <u>initial basis</u> S_o consists of the unit vectors $\{e_1, e_2, \ldots, e_m\}$, and each basis S_k in the sequence consists of certain unit vectors and certain vectors a_j of A. Reordering for convenience, we may suppose $S_k = \{a_1, \ldots, a_k, e_{k+1}, \ldots, e_m\}$. We write out the tableau of $A \cup B$ with respect to S_k as follows:

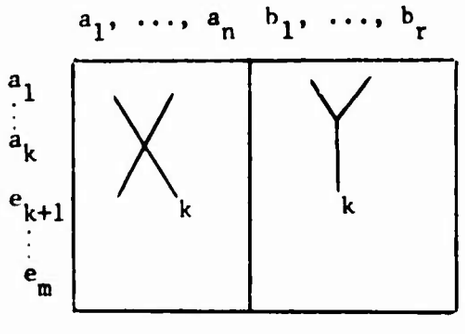


Figure 2.

where we denote the tableau of A and B with respect to S_k by X_k and Y_k respectively. There are two cases.

Case I: The last $m - k$ rows of X_k are zero. Then

(A) If the last $m - k$ rows of Y_k are also zero then Y_k is the desired solution of (I) since it expresses all the b_j linearly in $a_1, \ldots, a_k$.

(B) $y_{ij} \neq 0$ for some $i > k$. Then the problem has no solution; in fact, b_j is not a linear combination of the a_j. To see this note that the condition on X_k shows that the set $A_k = \{a_1, \ldots, a_k\}$ is a basis for A, but b_j is not a linear combination of A_k since the term $y_{ij} e_i$ occurs in the expression for b_j in terms of S_k. Hence the assertion follows:

Case II: $x_{ij} \neq 0$ for some $i > k$, say $i = k + 1$. Then let S_{k+1} be the basis obtained from S_k by <u>replacing</u> e_{k+1} by a_j. Thus $S_{k+1} = \{a_1, \ldots, a_k, a_j, e_{k+2}, \ldots, e_m\}$.

The proof that this algorithm solves Problem I is almost immediate. If Case II ever occurs then (A) the solution is either present or (B) it is seen not to exist. If Case I never occurs then after m replacements we will have constructed a basis S_m of vectors a_j from A and the tableau of B with respect to this basis is the desired solution.

Note that our method always produces a <u>basic solution</u>, i.e., a solution Y such that $y_{ij} \neq 0$ only for the basis $s_1, \ldots, s_n$. This proves the following fact which may not be immediately obvious:

THEOREM I: If (I) has a solution then it has a solution Y in which at least $n - m$ rows of Y are zero.

We now ask how much computation the replacement algorithm involves. Clearly

the only arithmetical step consists in going from a tableau with respect to a basis S to one with respect to S' obtained from S by replacing a single vector. In our present notation let Y and Y' be the tableaus of B with respect to S and S' and let the ith row of Y and Y' be denoted by y_i and y_i'.

THEOREM 2: Let $S = \{s_1, \ldots, s_m\}$ and suppose $y_{11} \neq 0$. Then $S' = \{b_1, s_2, \ldots, s_m\}$ is a basis and Y' is given by the rule

$$(1) \quad y_1' = y_1/y_{11}, \; y_i' = y_i - (y_{i1}/y_{11})y_1 .$$

Terminology: Operation (1) is known as pivoting and the element y_{11} is called the pivot element of the operation. It is easiest to rememeber the operation from these "pictures"

Y

	b_1	b_2,	...,	b_r
s_1	(y_{11})	y_{12},	...,	y_{1r}
s_2	y_{21}	y_{22},	...,	y_{2r}
$\vdots$				
s_m	y_{m1}	y_{m2},	...,	y_{mr}

Y'

	b_1	b_2,	...,	b_r
b_1	1	y_{12}',	...,	y_{1r}'
s_2	0	y_{22}',	...,	y_{2r}'
$\vdots$				
s_m	0	y_{m2}',	...,	y_{mr}'

Figure 4.

The pivot element is circled in Y. Pivoting is done by dividing the pivot row by the pivot element, and by adding a suitable multiple of the pivot row to each of the others, the suitable multiple being the one that will give zeros in the pivot column.

Proof of Theorem: Let Y' satisfy (1). Then $y'_{1j} = y_{1j}/y_{11}$

and $$y_{ij}' = y_{ij} - (y_{i1}/y_{11})y_{1j} \quad \text{so for } j \neq 1$$

$$y'_{1j}\, b_1 + \sum_{i=2}^{m} y'_{ij}\, s_i = y_{1j}/y_{11}\, b_1 + \sum_{i=2}^{m} y_{ij}\, s_i - (y_{1j}/y_{11}) \sum_{i=2}^{m} y_{i1}\, s_i$$

$$= y_{1j}/y_{11}\left(\sum_{i=1}^{m} y_{i1}\, s_i - \sum_{i=2}^{m} y_{i1}\, s_i\right) + \sum_{i=2}^{m} y_{ij}\, s_i = \sum_{i=1}^{m} y_{ij}\, s_i = b_j$$

so Y' is the tableau of B with respect to S'.

From rule (1) we see that each pivot step requires mr multiplications. For the matrix equation problem the number of columns of the tableau is $n + r$ (see Fig. 2.) and the problem is solved in at most m pivots so the number of multiplications is at most $m^2(n + r)$. Actually one does some what better than this because of the fact that after each pivot one gets columns which are unit vectors, like the first column of Y' in Figure 4.

Of special interest is the case where A and B are square $m \times m$ matrices and particularly where B is the identity matrix so that b_k is the kth unit vector. In this case X, if it exists, is A^{-1} and the pivot method involves exactly m^3 multiplications. Note that this is the number of multiplications used in multipling a pair of matrices, hence the number involved in checking a proposed solution X of $AX = I$. This suggests that m^3 multiplication is about as few as one could reasonably expect to use in solving the problem.

Finally note that we can follow the steps of the replacement algorithm even if there is no B matrix at all. The final tableau will then yield a column basis for A, and also, if one thinks about it for a moment, a proof that the row and column ranks of A are equal.

3. Solving Linear Inequalities

Problem II[†]: Given an $m \times n$ matrix A and an n-vector a find an m-vector y such that

$$y \geq 0 \quad \text{and} \quad yA \geq a .$$

It is convenient to rewrite the problem as follows: Find an m-vector y such that,

$$y\, a_j \geq \alpha_j \quad \text{for} \quad j = 1, \ldots, n$$
$$y\, e_i \geq 0 \quad \text{for} \quad i = 1, \ldots, m$$

where $\{e_i\}$ are the unit vectors of m-space.

Now there is no difficulty in finding a finite procedure for solving II, for it is easily shown, and will emerge from the procedure to be given here, that if II has a solution then it has a basic solution, that is a vector y such that $y\, a_j = \alpha_j$ and $y\, e_i = 0$ for some set of m vectors a_j and e_i which form a basis for m-space. One could, therefore, consider all bases among the vectors a_j, e_i and for each such compute the solution y to the corresponding m equations and then substitute this y into II. Eventually one of these vectors would satisfy the system unless there was no solution at all. Of course, this would be an enormously lengthy procedure since it would involve solving possibly $\binom{n+m}{m}$ systems of m equations in m unknowns.

We shall now describe a replacement algorithm for solving II. For this purpose we wish to transform II to a "homogeneous" problem, as follows:

[†]We are treating the case of nonnegative solutions of inequalities. The case in which y is unrestricted in sign can be handled in a similar way but involves some slight technical complication which we prefer to avoid in this exposition.

Let $\hat{a}_j$ be the $(m + 1)$-vector $(-\alpha_j, a_j)$ and let $\hat{e}_i = (0, e_i)$ for $i = 1, \ldots, m$ and let $e_0 = (1, 0, \ldots, 0)$ so that $e_0, \hat{e}_1, \ldots, \hat{e}_m$ are the unit vectors in $(m + 1)$-space. Finally let $\hat{Y}$ be all vectors $(1, y)$ where y is any m-vector.

Problem $\hat{II}$: Find y in $\hat{Y}$ such that

$$y \hat{a}_j \geq 0 \quad \text{for all} \quad j$$
$$y \hat{e}_k \geq 0 \quad \text{for all} \quad k .$$

It is clear from the definitions that Problems II and $\hat{II}$ are equivalent.

Now let $S = \{e_0, s_1, \ldots, s_m\}$ be a basis for $(m + 1)$-space where s_i is either a vector $\hat{a}_j$ or $\hat{e}_k$, and write the tableau with respect to this basis as follows:

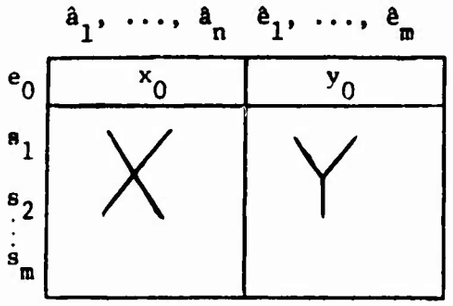


Figure 5.

THEOREM 3: If x_0 and y_0 are nonnegative then $(1, y_0)$ solves $\hat{II}$ (and y_0 solves II).

Proof: Let $\hat{y}$ be the $(m + 1)$-vector which solves the system

$$y s_i = 0 \quad i = 1, \ldots, m$$
$$y e_0 = 1$$

(this vector exists since S is a basis).

From the last equation above $\hat{y}$ is in $\hat{Y}$.

Now from the tableau we have

$$\hat{y}\ \hat{e}_k = y_{0k}(\hat{y}\ e_0) + \sum_{i=1}^{m} y_{ik}(\hat{y}\ s_i) = y_{0k}$$

so $\hat{y} = (1, y_0)$ and by assumption $y_0 \geq 0$. Finally

$$\hat{y}\ \hat{a}_j = x_{oj}(\hat{y}\ e_0) + \sum x_{ij}(\hat{y}\ s_i) = x_{oj} \geq 0$$

so $y = (1, y_0)$ solves $\hat{II}$, as asserted.

The inequality problem has now become that of finding a basis S so that the tableau of Figure 5. has its first row nonnegative, if such a basis exists, we wish to arrive at this basis by a sequence of replacements starting with the initial basis S_0 consisting of the unit vectors. The initial tableau is given below.

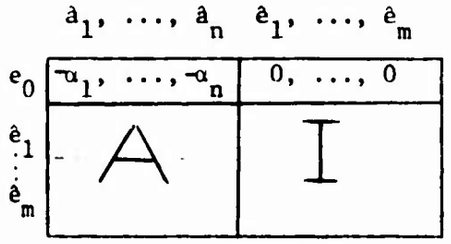


Figure 6.

Now, as in the previous section, we must describe the replacement operation. Suppose then that we have arrived at the tableau of Figure 5., but (x_0, y_0) is not positive so that say, $x_{0j} < 0$ (or $y_{0k} < 0$). Then by bringing $\hat{a}_j$ (or $\hat{e}_k$) into the next basis S' we can be sure that in the next tableau the entry x'_{0j} (or y'_{0k}) will be zero which would seem to be a step in the right direction.

The question which remains to be decided is which vector s_i in S should be replaced by $\hat{a}_j$ (or $\hat{e}_k$) and the success of the method depends on an ingenious criterion for making this decision which we now describe.

DEFINITION: An m-vector x is called <u>lexicographically positive</u>, or simply <u>ℓ-positive</u> if its first (reading from the left) nonzero coordinate is positive. We write

$$x \succ 0$$

A vector x is <u>lexicographically greater</u> than y, written $x \succ y$, if $x - y \succ 0$.

It is clear that for any $x \neq 0$ either $x \succ 0$ or $-x \succ 0$ so that $\succ$ defines a complete ordering of m-space with the further obvious property;

$$\text{if } x, y \succ 0 \quad \lambda, \mu \succ 0 \quad \text{then } \lambda x + \mu y \succ 0 .$$

Finally we call a matrix Y <u>ℓ-positive</u> if all of its rows are ℓ-positive. The following is the crucial notion for our algorithm.

DEFINITION: The basis S will be called <u>ℓ-feasible</u> if the matrix Y (Figure 5.) is ℓ-positive.

Note that the initial basis of Figure 6. in ℓ-feasible since in this case Y is the identity matrix. We now complete the description of the replacement algorithm. Assume in the tableau of Figure 5. that, say, x_{01} is negative (the argument would be the same for y_{01} negative). There are two cases.

Case I: The first column of X is nonpositive. Then we have

$$(2) \quad \hat{a}_1 = x_{01} e_0 + \sum_{i=1}^{m} x_{i1} s_i .$$

In this case II has no solution for if $\hat{y}$ solves II then $\hat{y}\, s_i \geq 0$ for all i, but then taking scalar product of (2) with $\hat{y}$ gives

$$\hat{y}\, \hat{a}_1 = x_{01} + \sum x_{i1}(\hat{y}\, s_i) \leq x_{01} < 0$$

so $\hat{y}$ cannot solve II.

Case II: $x_{i1} > 0$ for some i. Then let $I_1 = \{i | x_{i1} > 0\}$ and compute y_i/x_{i1} for i in I_1 and choose i_0 in I_1 for which y_i/x_{i1} is ℓ-minimal. Then obtain the new basis S' by replacing s_{i_0} by $\hat{a}_1$ (i.e., pivot on $x_{i_0 1}$).

The proof that this algorithm terminates depends on

LEMMA 2: The new basis S' is again ℓ-feasible and the vector y_0' of the new tableau is lexicographically greater than y_0.

Proof: From (1)

$$y'_{i_0} = y_{i_0}/x_{i_0 1} \quad \text{and since} \quad y_{i_0} > 0 \quad \text{and} \quad x_{i_0 1} > 0$$

it follows that $y'_{i_0} > 0$. Also for $i \neq i_0$

$$y'_i = y_i - (x_{i1}/x_{i_0 1})\, y_{i_0}.$$

If $x_{i1} \leq 0$ then clearly $y'_i > 0$. If $x_{i1} > 0$ then by the choice rule of Case II above

$$y_i/x_{i1} < y_{i_0}/x_{i_0 1}$$

Equality cannot hold here since this would mean that y_i and y_{i_0} were proportional which is impossible since Y is nonsingular, so again $y'_i > 0$ and hence Y' is ℓ-positive.

Also from (1)

$$y_0' = y_0 - \left(x_{01}/x_{i_0 1}\right) y_{i_0} \quad \text{and since}$$

x_{01} is negative, $x_{i_0 1}$ is positive and y_{i_0} is ℓ-positive we have $y_0' > y_{i_0}$ as asserted.

THEOREM 4: The replacement algorithm terminates.

Proof: Since the vector y_0 depends only on the basis S and since y_0 gets lexicographically larger at each iteration it is clear that no basis can recur. Therefore, one eventually arrives at the situation of Case I in which some column of the tableau is nonpositive in which case II has no solution, or else eventually (x_0, y_0) becomes nonnegative and y_0 is the desired solution.

4. An Example

Consider the system $y_1, y_2 \geq 0$

$$2y_1 + y_2 \geq 1$$
$$y_1 \geq 1$$
$$-y_2 \geq -1$$

The initial tableau is then

	$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{e}_1$	$\hat{e}_2$
e_0	-1	-1	1	0	0
e_1	2	1	0	1	0
e_2	(1)	0	-1	0	1

Now we will bring $\hat{a}_1$ into the next basis. According to the lexicographic rule $\hat{a}_1$ must replace e_2. The pivot element has been circled in the tableau above. The next tableau is

	$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{e}_1$	$\hat{e}_2$
e_0	0	-1	0	0	1
e_1	0	(1)	2	1	-2
$\hat{a}_1$	1	0	-1	0	1

The only possibility now is to replace $\hat{e}_1$ by $\hat{a}_2$ giving,

	$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{e}_1$	$\hat{a}_2$
e_0	0	0	2	1	-1
$\hat{a}_2$	0	1	2	1	-2
$\hat{a}_1$	1	0	-1	0	(1)

Again there is no choice. We must replace $\hat{a}_1$ by $\hat{e}_2$ and we get,

e_0	1	0	1	1	0
$\hat{a}_2$	2	1	0	1	0
$\hat{e}_2$	1	0	-1	0	1

which gives the solution $y_1 = 1$, $y_2 = 0$.

Note the way the row vector y_0 increases lexicographically with each replacement. Note too the interesting fact that the vector $\hat{e}_2$ was replaced on the first pivot step but came back in again in the end. Of course, if we had chosen to bring in $\hat{a}_2$ instead of $\hat{a}_1$ on the first replacement we would have obtained the solution in one step. However, in general there does not seem to be any good way of deciding which vector to bring in in order to minimize the number of replacements required to arrive at a solution.

5. Concluding Remarks

Having found an (apparently) good way to find at least one solution of a system of inequalities one might now ask for a way of finding all solutions, which means in essence finding all basic solutions of II. There do exist procedures for doing this but it is almost impossible to say whether these procedures are reasonable or not because of the fact that the number of basic solutions may increase very rapidly with m and n. The main interest here is theoretical. How many basic solutions can there be for an $m \times n$ system? I should like to conclude by describing very briefly the state of our knowledge (or ignorance) on this matter. For more details see Grünbaum [4].

It is conjectured that the maximum number μ of basic solutions which an $m \times n$ system can have is given by the strange looking formula

$$\mu(m, n) = 2\binom{n - \frac{m+1}{2}}{n - m} \qquad \text{for } m \text{ odd}$$

$$= \binom{n - \frac{m}{2}}{n - m} + \binom{n - \frac{m}{2} - 1}{n - m} \qquad \text{for } m \text{ even.}$$

This conjecture has in fact been proved for "most" values of m and n, specifically for all $m \leq 8$ and for $n \leq m + 3$ and $n \geq (m/2)^2 - 2$ to see what this means, the first unsolved cases are

$$m = 9 \quad 12 \leq n \leq 18 .$$

In general for each $m > 8$ there is an interval of values of n for which the conjecture has not been verified.

This strange situation together with the one described in the introduction concerning the number of replacements required to solve an $m \times n$ system are perhaps

the most interesting features of what might superficially appear to be a dull and routine problem. To mix metaphors a little, they indicate how close to the surface the so-called frontiers of mathematics sometimes lie.

Bibliography

[1] Dantzig, G. B., A. Orden, and P. Wolfe, "Generalized Simplex Method for Minimizing a Linear Form Under Linear Inequality Restraints," Pacific Journal of Mathematics, Vol. 5, pp. 183-195, (1955).

[2] Dantzig, G. B., "Constructive Proof of the Min-Max Theorem," Pacific Journal of Mathematics, Vol. 6, pp. 25-33, (1959).

[3] Debreu, Gerard, "Nonnegative Solutions of Linear Inequalities," International Economic Review, Vol. 5, No. 2, (May 1964).

[4] Grunbaum, B., CONVEX POLYTOPES, John Wiley & Sons, New York, (1967).

[5] Klee, V. L., "A Class of Linear Programming Problems Requiring a Large Number of Iterations," Numerische Mathematik, Vol. 7, pp. 313-321, (1965).

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY *(Corporate author)*

University of California, Berkeley

2a REPORT SECURITY CLASSIFICATION

Unclassified

2b GROUP

3 REPORT TITLE

HOW TO SOLVE LINEAR INEQUALITIES

4 DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Research Report

5 AUTHOR(S) *(Last name, first name, initial)*

GALE, David

6 REPORT DATE	7a TOTAL NO OF PAGES	7b NO OF REFS
July 1967	17	5

8a CONTRACT OR GRANT NO	9a ORIGINATOR'S REPORT NUMBER(S)
Nonr-222(83) b PROJECT NO NR 047 033 c Research Project No.: RR 003 07 01 d	ORC 67-26 9b OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10 AVAILABILITY/LIMITATION NOTICES

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED.

11 SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY
Also supported by the National Science Foundation under Grant GP-7417.	MATHEMATICAL SCIENCE DIVISION

13 ABSTRACT

This paper is an attempt to solve the following constrained minimum problem: To present the most easily described algorithm for solving linear inequalities subject to the constraints,

(1) The algorithm must be efficient.
(2) It must be shown to terminate.

The algorithm is a variant of the lexicographic simplex method which avoids using any artificial objective function.

DD FORM 1 JAN 64 1473

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Lexicographic Simplex Method						
Pivotal Methods						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, &, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.